

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

(підпис) Тарасенко В.П.
(ініціали, прізвище)

“ ____ ” червня 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра**

з напрямку підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: «Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах».

Виконав: студент IV курсу, групи KB-51

Гурський Олександр Сергійович

(підпис)

Керівник, проф. каф. СПіСКС, д.т.н., проф. Терейковський І.А.

(підпис)

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М.

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Тарасенко В.П.
(ініціали, прізвище)

«__» червня 2019 р.

**ЗАВДАННЯ
на дипломний проект студента
Гурського Олександра Сергійовича**

1. Тема проекту: «Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах»,
керівник проекту: проф. каф. СПіКСС, д.т.н., проф. Терейковський І.А.,
затверджені наказом по університету від «22»травня 2019 р. №1330-С
2. Термін подання студентом проекту _____
3. Вихідні дані до проекту: див. технічне завдання.
4. Зміст пояснювальної записки: аналіз існуючих рішень та обґрунтування теми дипломного проекту, методи використання нейронних мереж для розпізнавання деструктивного контенту, розробка інструментальних засобів, аналіз розробленої системи.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): структурні схеми - навчання нейронної мережі

для аналізу тональності тексту бази даних, класифікація нового тексту навченою мережею, процедура попередньої обробки текстів, архітектура LSTM для класифікації тексту.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, к.т.н.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Вивчення літератури за тематикою роботи	15.02.2019
2.	Розробка технічного завдання	12.03.2019
3.	Аналіз існуючих рішень	14.04.2019
4.	Розробка структури та алгоритмів системи	29.04.2019
5.	Аналіз розробленої системи	10.05.2019
6.	Підготовка матеріалів пояснювальної записки	14.05.2019
7.	Підготовка матеріалів графічної частини проекту	20.05.2019
8.	Оформлення документації дипломного проекту	26.05.2019

Студент

(підпис)

Гурський О.С.
(ініціали, прізвище)

Керівник проекту

(підпис)

Терейковський І.А.
(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (54 с., 21 рис. 5 табл., 2 додатки).

Об'єкт розробки – створення нейромережевої системи, яка дозволяє розпізнавати деструктивний контент в текстових веб-ресурсах.

Система дозволяє: аналізувати тональність тексту з метою пошуку деструктивного контенту. Була вибрана бінарна шкала оцінювання: позитивна та негативна оцінки. Обраним типом мережі стала рекурентна нейронна мережа. В якості бази даних використовувалась база даних відкритого доступу.

В ході розробки:

- проведено аналіз методів класифікації тональності тексту;
- сформульовані вимоги до нейромережевої системи розпізнавання деструктивного контенту;
- розроблена система розпізнавання деструктивного контенту;
- розроблена структура нейронної мережі для класифікації тексту;
- розроблено програмне забезпечення для розпізнавання деструктивного контенту в веб-ресурсах на основі аналізу тональності тексту.

Упровадження цієї системи в певні веб-ресурси дозволить оцінити думки користувачів, знайти негативні відгуки, що дасть змогу проаналізувати потенційні недоліки досліджуваного об'єкту.

Ключові слова:

СИСТЕМА РОЗПІЗНАВАННЯ ДЕСТРУКТИВНОГО КОНТЕНТУ, АНАЛІЗ ТОНАЛЬНОСТІ ТЕКСТУ, ВЕБ-РЕСУРСИ, БІНАРНА ШКАЛА, РЕКУРЕНТА НЕЙРОННА МЕРЕЖА.

ABSTRACT

The diploma project includes an explanatory note (54 p., 21 fig., 2 appendices).

The object of development - the creation of a neural network system, which allows you to recognize destructive content in text web resources.

The system allows you to: analyze the tone of the text in order to find destructive content. The binary rating scale was selected: positive and negative. The chosen type of network has become a recurrent neural network. The database used an open access database.

In the development process were resolved:

- analysis of methods of classification of tonality of the text;
- requirements for neural network detecting of destructive content are formulated;
- a destructive content recognition system developed;
- the structure of the neural network is developed for the classification of the text;
- software for detecting destructive content in web resources based on the analysis of the tone of the text is developed.

Implementation of this system in certain web resources will allow to evaluate the opinions of users, to find negative reviews, which will allow to analyze the potential disadvantages of the investigated object.

Keywords:

DESTRUCTIVE CONTENT DETECTING SYSTEM, TONE ANALYSIS OF THE TEXT, WEB RESOURCES, BINARY RATING SCALE, RECURRENT NEURAL NETWORK.

[illegible]

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1. Вимоги до програмного продукту, що розробляється	3
5.2. Вимоги до апаратного забезпечення	3
5.3. Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045420.002 ТЗ					
Змін	Арк.	№ докум.	Підпис	Дата						
Розробив		Гурський О.С.			Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах Технічне завдання			Літ.	Аркуш	Аркушів
Перевірив		Терейковский І.А.							1	4
								КПІ ім. Ігоря Сікорського, ФПМ КВ-51		
Н. контроль		Клятченко Я.М.								
Затвердив		Тарасенко В.П.								

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах».

Галузь застосування: веб-ресурси, що містять текстовий контент, який потрібно проаналізувати та сформулювати необхідні висновки щодо певного об'єкта аналізу.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є розробка нейромережевої системи розпізнавання деструктивного контенту в текстових веб-ресурсах для аналізу емоційного забарвлення тексту та оцінки його деструктивності.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

					ІАЛЦ.045420.002 ТЗ	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- Класифікація нового тексту та виведення оцінки щодо його деструктивності;
- Наявність модуля попередньої обробки даних для подання їх у необхідному вигляді;
- Можливість змінювати вхідні параметри нейронної мережі;
- Можливість перенавчання нейронної мережі для збільшення якості її класифікації.

5.2. Вимоги до апаратного забезпечення

- оперативна пам'ять: від 8 Гб;
- процесор: CPU частотой від 2.4 ГГц, 4 або більше ядер або GPU

5.3. Вимоги до програмного та апаратного забезпечення користувача

- операційна система Linux/Windows;
- наявність доступу до мережі інтернет.
- наявність Google диску, браузера та аккаунта.

					ІАЛЦ.045420.002 ТЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою роботи	15.02.2019
2.	Розробка технічного завдання	12.03.2019
3.	Аналіз існуючих рішень	14.04.2019
4.	Розробка структури та алгоритмів системи	29.04.2019
5.	Аналіз розробленої системи	10.05.2019
6.	Підготовка матеріалів пояснювальної записки	14.05.2019
7.	Підготовка матеріалів графічної частини проекту	20.05.2019
8.	Оформлення документації дипломного проекту	26.05.2019

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.002 ТЗ

Арк.

4

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.004 ПЗ	Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах.	54		
			Пояснювальна записка			
	A4	ІАЛЦ.045420.005 Д1	Навчання нейронної мережі для аналізу тональності тексту.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.006 Д2	Класифікація нового тексту навченою мережею.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.007 Д3	Процедура попередньої обробки текстів.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.008 Д4	Архітектура LSTM для класифікації тексту.	1		
			Схема структурна			

[illegible]

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	3
ВСТУП.....	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ.....	5
1.1 Загальна характеристика проблеми розпізнавання деструктивного контенту	5
1.2 Аналіз існуючих рішень розпізнавання деструктивного контенту в текстових веб-ресурсах	8
1.3 Постановка задач дослідження.....	15
2. МЕТОДИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ДЕСТРУКТИВНОГО КОНТЕНТУ	17
2.1 Визначення перспективних нейромережових архітектур.....	17
2.2 Структура нейронних мереж	20
2.3 Використання рекурентних нейронних мереж	28
3. РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ	42
3.1 Модель рекурентної нейромережі для класифікації тексту	42
3.2 Програмні засоби для роботи системи	43
3.2.1 Попередня обробка даних	44
4. АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ	46
4.1 Технологія використання	46
4.2 Експериментальні дослідження	48
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	52
ДОДАТКИ	

Додаток 1. Копії графічних матеріалів

					ІАЛЦ.045420.004 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата	Нейромережева система розпізнавання деструктивного контенту в текстових веб-ресурсах	Літ.	Аркуш	Аркушів
Розробив		Гурський О.С.					1	54
Перевірив		Терейковський І. А.						
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-51		
Затвердив		Тарасенко В. П.						
					Пояснювальна записка			

ІАЛЦ.045420.005 Д1. Навчання нейронної мережі для аналізу тональності тексту. Схема структурна

ІАЛЦ.045420.006 Д2. Класифікація нового тексту навченою мережею. Схема структурна

ІАЛЦ.045420.007 Д3. Процедура попередньої обробки текстів. Схема структурна

ІАЛЦ.045420.008 Д4. Архітектура LSTM для класифікації тексту. Схема структурна

Додаток 2. Презентація

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		2

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

SA – Sentiment Analysis (аналіз тональності);
CNN – Convolutional Neural Network (згортова нейронна мережа);
RNN – Recurrent Neural Networks (рекурентна нейронна мережа);
LTSM – Long Short-term Memory (мережа довгої короткочасної пам'яті);
CEO – Correctly Extracted Opinions (правильно розпізнані думки);
TNOFS – Total Number of Opinions Found by System (загальна кількість думок знайдених системою);
GRU – Gated Recurrent Unit (Керований рекурентний блок);
НМ – Нейронна мережа;
КС – Комп'ютерна система;
GPU – Graphics Processing Unit (графічний процесор);
CPU – Central Processing Unit (центральний процесор);
Embedding – шар нейронної мережі;
Bag of words – спосіб векторного подання слів;
Adam – Adaptive moment estimation.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		3

ВСТУП

У зв'язку з розвитком технологій інтернет ресурси стали невід'ємною частиною життя людини. Через наявність різноманітних веб-сервісів, в яких можна висловити своє ставлення до певного об'єкта – збільшилася кількість суб'єктивного контенту: повідомлення в блогах та соціальних мережах, відгуки щодо послуг чи товарів, думки, виражені стосовно фільмів, книг, ресторанів чи політики. Весь цей контент створений інтернет користувачами та спільно ними ж і використовується. Суб'єктивний контент являє собою цінний ресурс. Адже це величезна кількість неструктурованої інформації, яку можна використовувати. Але кількість відгуків, що публікуються в тих же соціальних мережах, може досягати десятків тисяч. Їх обробка повна людиною стає неможливою.

Деструктивний контент це певний вид суб'єктивного контенту, що має негативне емоційне забарвлення. Розпізнавання таких текстів, можливо, з майбутнім видаленням є перспективним завданням, так як думки інших людей часто впливають на рішення людини в будь-яких ситуаціях. Так, вибираючи товар в інтернет-магазині, фільм на вечір, книгу користувачі віддають перевагу об'єкту, який набрав якомога більше позитивних відгуків.

Читаючи відгуки покупців, наприклад, юридична особа торгової мережі, може дізнатися ставлення клієнтів до процесу обслуговування в конкретних відділах. Їх дослідження здатне допомогти виявити раніше невідомі дефекти товару. Тому думки людей представляють величезний інтерес для соціологів, маркетологів, власників бізнесу. Відгуки можуть бути корисні і пошуковим системам, результати їх аналізу дозволять виключення з результатів пошуку об'єкти, які мають негативне емоційне забарвлення. Все це робить актуальним завдання аналізу тональності текстів.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		4

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Загальна характеристика проблеми розпізнавання деструктивного контенту

Кожного дня в веб-ресурсах з'являється чисельна кількість нового контенту: користувачі доносять свою думки про фільми, спорт, новини, залишають відгуки стосовно різноманітних товарів, послуг та інших об'єктів. Іноді їх слова можуть мати певну деструктивну спрямованість, тобто нести за собою негативні, агресивні чи насильницькі дії або закликають до їх вчинення. Ключовою ознакою деструктивної спрямованості тексту є тема руйнування конкретної соціальної структури, її елементів, властивостей, характеристик, відносин з іншими структурами. Зазвичай, він містить надмірну кількість емоційно-риторичних елементів, щоб спростити думку суб'єкта для читача [1].

Для вирішення завдань, пов'язаних з аналізом емоційного забарвлення тексту, використовуються методи аналізу тональності тексту.

Sentiment analysis (аналіз тональності) - це область комп'ютерної лінгвістики, яка займається вивченням думок і емоцій авторів по відношенню до об'єктів, які розглянуті в тексті.

Метою SA є знаходження певних роздумів в тексті та визначення їх властивостей. Залежно від поставленого завдання, необхідними можуть бути різні властивості, а саме:

автор – той, хто написав свою думку;

тема - про що саме було написано;

тональність - відношення автора щодо згаданої теми, що зазвичай є позитивним чи негативним.

У літературі зустрічаються різні способи представлення моделі

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		5

мислення. Також використовується і різна термінологія. В англійській мові цю область дослідження зазвичай називають *opinion mining and sentiment analysis* (пошук думок і аналіз почуттів). У українських статтях зазвичай вживається термін - аналіз тональності. Незважаючи на те, що тональність є лише однією з характеристик, саме завдання її класифікації є найбільш часто досліджуваним в наші дні. Це можна пояснити кількома причинами:

1. Визначення автора і теми є більш складним завданням ніж класифікація тональності, тому є сенс спочатку вирішити більш просту задачу, а пізніше переключитися на інші.
2. У багатьох випадках потрібно визначити лише тональність, тому що інші характеристики нам уже відомі. Наприклад, при зборі відгуків з блогів, зазвичай думка належить автору посту, тобто визначати автора вже не потрібно. Також часто відома тема: наприклад, якщо відбувається пошук за ключовими словами певного тексту, то потрібно лише визначити його тональність. Звичайно ж, це працює не у всіх випадках, а тільки в більшості з них. Але ці припущення дозволяють в значній мірі спростити і так нелегке завдання [2].

Основним завданням в аналізі тональності є визначення полярності даного тексту, тобто, чи є виражені думки позитивними, негативними або нейтральними. Існують різні способи визначення полярності даного матеріалу:

1. За бінарною шкалою - для визначення полярності написаного використовується два класи оцінок: позитивна та негативна. Одним з недоліків цього підходу є те, що емоційну складову не завжди можна однозначно визначити, тобто текст може містити ознаки позитивної оцінки, так і негативної.
2. За багатосмуговою шкалою, що розроблена Пангом [3]. Ним було

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		6

розширене основне завдання класифікації кіновідгуків від оцінки за бінарною шкалою до прогнозування рейтингу по 3-х або 4-бальною шкалою.

3. Використання систем шкалювання, за допомогою яких, словам, пов'язаних з негативною, нейтральною чи позитивною тональностями, ставляться відповідно числа за шкалою від -10 до 10 (від негативного до самого позитивного).

Одна із проблем визначення емоційної оцінки тексту полягає в тому, що сама задача визначення - суб'єктивна. Так, згідно з деякими дослідженнями [4], різні люди можуть по-різному оцінити один і той же текст. Більш того, думки можуть бути протилежними - частина людей відносить текст в позитивний клас, а частина - в негативний. Самі тексти на природній мові є неструктурованими об'єктами, що ускладнює роботу з ними. Крім того, в тексті можуть бути присутніми сарказм, жарти, граматичні помилки, які не завжди може зрозуміти людина, не кажучи вже про машину.

Ще однією проблемою є те, що розроблені алгоритми можуть працювати для конкретної мови і не працювати для інших. Наприклад, програма, що розробляється для текстів, написаних англійською мовою, може не працювати для написаних українською чи російською. Також тональність тексту безпосередньо залежить від предметної області. Завдяки тому, що емоційна оцінка одного і того ж слова може змінюватися в різних предметних областях. Так, слово нейтральне в одній області може бути оціночним словом в іншій.

Звичайно, слово може приймати також різну тональність і залежно від контексту.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		7

1.2 Аналіз існуючих рішень розпізнавання деструктивного контенту в веб-ресурсах

Перше, що необхідно для розпізнавання деструктивного контенту - певна база даних (відгуків, текстів) для аналізу. Розпізнавання деструктивного контенту є складним завданням, так як його конкретного визначення немає. Більшість ресурсів вказує на те, що це – емоційно забарвлений текст, який несе в собі думки індивіда, сенсом яких є вчинення агресивних чи негативних дій. Тому існує термін тональності тексту, визначення якої і є рішенням для ідентифікації деструктивного контенту.

Комп'ютери можуть виконувати аналіз цифрових текстів, тільки використовуючи елементи машинного навчання, а саме: прихований семантичний аналіз, метод опорних векторів, “мішок слів” та інші. В більш складних методах зазвичай визначається власник з даними емоціями (певна людина) і ціль (тобто об'єкт, щодо якого виражаються почуття). Щоб визначити думку без вилучення існуючого контексту, використовують граматичні зв'язки між словами [3].

Граматичні зв'язки визначаються на основі повного структурного розбору тексту. Аналіз тональності може бути розділений на дві окремі категорії:

- ручний (аналіз тональності експертами);
- автоматизований аналіз тональності.

Найбільш помітні відмінності між ними лежать в ефективності системи і точності аналізу. У комп'ютерних програмах автоматизованого аналізу тональності застосовують алгоритми машинного навчання, інструменти статистики та обробки природньої мови, що дозволяє обробляти великі масиви тексту, включаючи веб-сторінки, онлайн-новини, тексти груп для спілкування в мережі Інтернет, рецензії на певні об'єкти,

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		8

веб-блоги тощо.

Аналіз тональності зазвичай відносять до однієї з задач комп'ютерної лінгвістики, тобто мається на увазі, що можливо знайти і класифікувати тональність, використовуючи інструменти обробки природної мови. Існують різні методи для класифікації тональності [2]. Найбільш відомі з них:

- методи, засновані на правилах;
- методи, засновані на словниках;
- машинне навчання зі вчителів;
- машинне навчання без вчителя;
- метод, заснований на теоретико-графових моделях.

1. Методи, засновані на правилах

Цей підхід шукає думки в тексті і класифікує їх, базуючись на кількості негативних і позитивних слів. У ньому можуть бути різні правила класифікації, такі як тональні словники, суперечливі слова, ідіоми, слова які підсилюють значення, змішані думки та інші. Самі правила будуються на основі шаблонів, виділених з тексту, які часто зустрічаються. Це можуть бути слова і фрази. Даний тип систем, що складається з певної кількості правил, використовує їх, та показує результат щодо тональності тексту.

Наприклад, маємо речення - “я обожаю морозиво”. Для нього можна використати таке правило:

Якщо присудок “обожнюю” входить в позитивний набір дієслів: “кохаю, обожаю, схвалюю” та в реченні немає заперечень, то можна класифікувати тональність як позитивна.

Багато комерційних систем використовують даний підхід, незважаючи на те що він вимагає великих витрат, тому що для хорошої роботи системи необхідно скласти велику кількість правил. Найчастіше

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		9

правила прив'язані до певної тематики (наприклад, “кіно”) і при її зміні (“спорт”) потрібно заново складати правила. Тим не менш, цей підхід є найбільш точним при наявності хорошої бази правил, але зовсім не цікавим для дослідження.

2. Методи, засновані на словниках

Такі системи використовують так звані *affective lexicons* (тональні словники) для аналізу тексту. У простому вигляді в тональному словнику міститься список слів зі значенням тональності для кожного слова. Розглянемо приклад.

Таблиця 1.1

Приклад фрагменту тонального словника

Слово	Тональність(1-10)
щасливий	9.5
позитивний	8
нудний	3
сердитий	2.7
сумний	1.9

Для аналізу цього тексту, можна користуватися наступним алгоритмом:

1. Кожному слову в тексті присвоїти його значення тональності зі словника (якщо воно присутнє в словнику).
2. Обчислити загальну тональність всього тексту.

Останній пункт можна зробити різними способами. Найпростіший з них – обчислити середнє арифметичне значень тональностей усіх слів в реченні. Більш складний – навчання класифікатора, наприклад, нейронної мережі.

Даний метод вимагає значних трудовитрат, так як для хорошої роботи системи потрібна велика база слів з їх вагами (тональностями). Проте існують способи, що дозволяють автоматизувати складання словників для конкретної предметної області.

3. Машинне навчання без вчителя

В основі цього методу лежить така ідея: спочатку знаходяться терміни, які найчастіше зустрічаються в даному тексті, і, водночас, присутні в незначній кількості в інших текстах даної бази. В такій системі ці слова мають найбільшу вагу в джерелі інформації. Виділяючи їх, після визначення тональності кожного такого терміну, виконується аналіз щодо тональності всього тексту [4]. Одним з прикладів такої системи може бути автоматична кластеризація документів.

4. Метод, заснований на теоретико-графових моделях

Основою цього методу є припущення, що не всі слова в певному текстовому масиві даного документа є рівнозначними. Якісь з них мають більшу вагу, тим самим сильніше впливають на тональність тексту. При використанні такого способу аналіз тональності розбивається на кілька етапів:

1. побудова графа на основі досліджуваного тексту;
2. ранжування його вершин;
3. класифікація знайдених слів;
4. обчислення результату.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		11

Для класифікації слів використовується тональний словник, в якому кожному слову співвідноситься оцінка, наприклад “позитивна”, “негативна” чи “нейтральна”. Щоб отримати кінцевий результат потрібно обчислити значення двох частин: позитивної складової тексту і негативної. Для того, щоб знайти позитивну частину тексту необхідно знайти суму тональностей всіх позитивних термінів джерела даних з урахуванням їх ваги. Значення негативної складової знаходиться аналогічним чином. Для підсумкової оцінки тональності всього тексту потрібно обчислити відношення цих складових за формулою:

$$T = P / N ,$$

де Т - підсумкова оцінка тональності, Р - оцінка позитивної складової тексту і N - негативна складова тексту. Відповідно до статті Меншикова [5]: текст, в якому значення Т близьке до одиниці, буде вважатися нейтральним, якщо трохи перевищує одиницю – позитивним, сильно перевершує - дуже позитивним. Вірно і зворотнє для текстів негативної тональності.

5. Машинне навчання зі вчителем

Методи на основі машинного навчання з учителем є найбільш популярними в дослідженнях [2]. Основний принцип полягає в тому, що машинний класифікатор спочатку навчається за допомогою готової бази даних оброблених текстів, а потім отримана модель поведінки використовується для аналізу нових документів. Навчальні дані містять у собі приклади та ймовірні відповіді на них. Якщо розроблена система успішно працює, то при подачі на вхід нових даних, вони будуть зіставлені з правильними відповідями.

Метод навчання зі вчителем буває двох типів: класифікація та регресія.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		12

У випадку класифікації визначається кількість класів, множина яких складає собою кількість видів тональності. Наприклад, при бінарній класифікації - позитивна та негативна тональності. При трьохкласовій до двох попередніх додається ще й нейтральна.

Ціллю метода з регресією є навчання класифікатора для отримання чисельного значення, наприклад від 1 до 10, де більше значення означає більш позитивну тональність.

Алгоритм створення системи:

1. Збирається необхідна база даних документів, за допомогою якої навчається машинний класифікатор;
2. Кожен документ представляється у вигляді вектора ознак, за якими він буде досліджуватися;
3. Вказується правильна відповідь (тип тональності) для кожного елемента бази;
4. Вибирається алгоритм класифікації та метод для навчання системи;
5. Проходить процес навчання;
6. Використовуємо отриману модель для визначення тональності текстів нової бази даних.

При створенні нейронної мережі, користуючись даним методом з ціллю аналізу тональності тексту, відомі використання двох типів мереж: CNN (convolutional neural network) та RNN (recurrent neural networks) в парі з модулем LTSM (long short-term memory).

В цілому, немає оптимального рішення, яке б підійшло для будь якого поставленого завдання щодо аналізу тональності тексту. Кожен з

методів класифікації тональності має свої переваги та недоліки. Розглянемо основні з них.

Таблиця 1.2

Порівняльна характеристика методів класифікації тональності

метод, заснований на правилах	метод, заснований на словниках	машинне навчання без вчителя	машинне навчання зі вчителем	на теоретико- графових моделях
переваги				
висока точність	простий в використанні	автоматичний	автоматичний	висока точність
використовується в комерційних системах		не потребує даних для навчання		
недоліки				
витратний	не універсальний	низька точність	потребує дані для навчання	Складний у реалізації

Важливим завдання в розробці системи є оцінка її якості роботи. Точність та якість системи аналізу тональності тексту оцінюється тим, наскільки добре її відповідь співпадає з думкою людини стосовно емоційної оцінки досліджуваного тексту. Для цього можуть використовуватися такі параметри як точність та повнота [6]. Формула для знаходження повноти:

$$R = \frac{CEO}{TNO},$$

де CEO – correctly extracted opinions (правильно розпізнані думки), TNO – total number of opinions (загальна кількість думок, як знайдених системою, так і не знайдених).

Точність обчислюється за формулою:

$$R = \frac{CEO}{TNOFS},$$

де CEO – правильно розпізнані думки, TNOFS – total number of opinions found by system (загальна кількість думок знайдених системою). Отже, точність виражає кількість досліджуваних текстів, речень чи документів, в оцінці яких, думка системи SA збіглася з думкою експерта. При цьому, згідно з дослідженнями [7], експерти зазвичай погоджуються в оцінках тональності конкретного тексту в 79 % випадків. Тобто, програма, яка визначає тональність тексту з точністю 70 % робить це майже так само добре, як і людина.

1.3 Постановка задач дослідження

Метою даної роботи є розробка нейромережевої системи для розпізнавання деструктивного контенту в текстових веб-ресурсах. Для цього потрібно вивчити на базі яких алгоритмів можливо виконувати дану роботу та розглянути відомі методи застосування нейронних мереж у цій сфері. Це потребує ознайомлення з роботою алгоритмів аналізу тональності тексту та визначення перспективних нейромережевих архітектур в цій галузі. Після вибору архітектури нейронної мережі потрібно детально вивчити особливості її використання.

Далі необхідно знайти оптимальний набір вхідних даних нейронної мережі, який включає в себе різні параметри.

Також важливими кроками у розробці системи є створення алгоритму навчання та використання нейронної мережі.

Можна сформулювати такі вимоги до нейромережевої системи розпізнавання деструктивного контенту в текстових веб-ресурсах:

- слід застосувати адаптовані до поставленої задачі розпізнавання сучасні види рекурентних нейронних мереж;
- програмний додаток повинен відповідати сучасним вимогам до надійності, швидкості та ресурсоемності;
- програмний додаток повинен бути орієнтований на використання в комп'ютерних системах, що функціонують під управлінням сучасних операційних систем.

Крім того, проведений аналіз задачі розробки захисту інформації в офісному документообігу дозволяє виділити такі етапи її вирішення:

1. Побудова архітектури системи розпізнавання деструктивного контенту.
2. Розробка математичного забезпечення комп'ютерної системи.
3. Розробка інформаційного забезпечення.
4. Розробка програмного забезпечення.
5. Експериментальне підтвердження отриманих результатів.

2. МЕТОДИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ДЕСТРУКТИВНОГО КОНТЕНТУ

2.1 Визначення перспективних нейромережових архітектур

Якщо провести аналіз стану сучасних нейромережових технологій, то можна сформулювати висновок про те, що доцільність застосування конкретного типу НМ необхідно визначати на основі співставлення характеристик мережі з умовами поставленої задачі. До таких характеристик та умов відносяться:

- вимоги до обчислювальних потужностей;
- параметри навчальних даних;
- обмеження технічної реалізації НМ;
- загальні обмеження процесу навчання;
- вимоги до вихідної інформації,
- сфера застосування.

Розглянемо вказані характеристики.

1. На практиці вимоги до обчислювальних потужностей визначаються максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. В свою чергу достовірність прийняття рішення характеризується допустимими величинами максимальної та середньої помилки мережі на реальних даних які в загальному випадку можуть виходити за межі множини навчальних даних. Відповідно виникає задача екстраполяції результатів навчання НМ за межі навчальних прикладів. Відзначимо, що обчислювальна потужність мережі залежить від її типу та алгоритму навчання. Ще однією вимогою може бути незмінність виходу мережі для різних прикладів з однаковими параметрами.

2. До основних параметрів навчальних даних відносяться:

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		17

- Кількість параметрів, що характеризують навчальний приклад.
- Вид параметрів, дискретний (символьний) чи безперервний (числовий).
- Загальна кількість навчальних прикладів.
- Наявність помилок (шуму) в навчальних прикладах.
- Наявність кореляції навчальних прикладів.
- Можливість та необхідність попередньої обробки вхідних даних з метою їх нормалізації та видалення шуму.
- Повнота виборки, тобто можливість відображення в ній всіх аспектів процесу, що моделюється. Наприклад, чи можливо відобразити в навчальній вибірці сигнатури всіх вірусів, або сигнатури мережових атак певного типу.
- Пропорційність навчальних прикладів, що відповідають різним аспектам процесу, що моделюється. Наприклад скільки навчальних прикладів відповідають мережній атаці типу А, а скільки прикладів – атаці типу В.

3. Загальні обмеження процесу навчання обумовлюються:

- Максимальним терміном навчання.
- Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Цим визначається тип навчання – з вчителем або без вчителя.
- Можливістю автоматизації процесу навчання, яка визначається кількістю та важливістю емпіричних параметрів. Вказана можливість багато в чому визначає умови застосування НМ. Мережі в яких процес навчання не автоматизовано можуть використовуватись тільки в лабораторних умовах.
- Можливістю донавчання в процесі експлуатації.

– Вимогами до якості навчання, яке звичайно оцінюють по величині максимальної та середньої помилки розпізнавання навчальних та тестових даних. При цьому тестові дані повинні не значно відрізнятись від навчальних.

– Можливістю навчання НМ в лабораторних умовах. Наприклад, в лабораторних умовах потенційно можливо навчити НМ розпізнавати мережеві атаки певного типу. В той же час неможливо навчити НМ класифікувати електронні листи відповідно інтересам конкретного користувача. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

4. Обмеження технічної реалізації НМ стосуються: швидкості прийняття рішення, інтеграції в існуючі системи, обсягу та складності програмної реалізації. Для зменшення обсягу можливо розділити програмний код для навчання мережі від коду, що відповідає за її функціонування.

5. Вимоги до вихідної інформації НМ вказують на те в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнаванні вірусів може виникнути необхідність не тільки визначення ситуації “вірус А присутній”, але й розрахунку ймовірності цієї ситуації. Стосовно класифікації електронних листів вихідною інформацією НМ може бути відображення листів на площину, яке дозволить провести остаточну класифікацію користувачеві. Ще однією вимогою може бути необхідність визначення вербальних залежностей між вхідною та вихідною інформацією.

6. Сфера застосування визначає системи в яких буде використовуватись НМ. На сьогодні достатньо дослідженим є

використання НМ для розпізнавання образів та при проведенні оптимізаційних розрахунків. Відзначимо, що системи розпізнавання образів принципово відрізняються від систем аналізу тексту тим, що в них кількість вихідних та кількість комбінацій вхідних параметрів принципово обмежена. В системах аналізу тексту ця кількість принципово необмежена. В системах захисту від спаму можливо використати НМ призначені для аналізу тексту. В системах керування слід застосувати НМ призначені для проведення оптимізаційних розрахунків. В перспективі доцільно застосувати НМ з метою реалізації паралельних розрахунків в КС, що дозволить значно підвищити їх стійкість від багатьох типів атак з метою відмови в обслуговуванні. Крім того сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ повинно бути передбачено можливість повної автоматизації процесу донавчання на експлуатації.

Аналізуючи дані характеристики можна визначити принципову доцільність застосування того чи іншого типу НМ для вирішення задачі. Остаточне рішення про використання конкретного типу НМ із декількох можливих повинно бути прийняте після проведення порівняльних експериментів.

2.2 Структура нейронних мереж

Штучні нейронні мережі були побудовані за принципом біологічних, що представляють собою мережі нервових клітин, які виконують певні фізіологічні функції. У загальному вигляді вона являє собою математичну обчислювальну модель для введення даних (сигналів) у вхідний шар нейронів, їх обробки в прихованих шарах і отримання вихідного шару. Ключова особливість системи в тому, що вона здатна до самонавчання і не є явно запрограмованою.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		20

Нейронна мережа складається з великої кількості нейронів, які зберігають в собі певні значення коефіцієнтів та мають певну вагу.

Нейрон виконує декілька функцій:

- приймальна: синапси отримують інформацію;
- інтеграційна: на виході нейрона сигнал, який несе інформацію про всі підсумовування всіх сигналів в нейроні;
- провідникова: по аксону проходить інформація до синапсу;
- функція передачі: передається певний імпульс від одного нейрона до іншого.

Синапсами називають зв'язки, за допомогою яких вихідні сигнали одних нейронів надходять на входи інших. Кожна зв'язок характеризується своєю вагою. Зв'язки з позитивним вагою називаються збудливими, а з негативним - гальмуючими. Вихід нейрона називається аксоном. У штучній нейронній мережі, нейрон - це деяка нелінійна функція, аргументом якої є лінійна комбінація всіх вхідних сигналів. Така функція називається активаційною. Потім результат активаційної функції посиляється на аксон. Об'єднуючи такі нейрони, отримують штучну нейронну мережу.

Функція активації нейрона характеризує залежність сигналу на виході нейрона від суми сигналів на його входах. Зазвичай функція є монотонно зростаючою і знаходиться в області значень $[-1,1]$ (гіперболічний тангенс) або $[0,1]$ (сигмоїд). Для деяких алгоритмів навчання необхідно, щоб вона була безперервно диференційною на всій числовій осі. Штучний нейрон характеризується своєю активаційною функцією (наприклад, назва "сигмоїдний нейрон").

Основними активаційними функціями є:

- порогова активаційна функція (функція Хевісайда).

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		21

$$f(x) = \begin{cases} 1 & \text{if } x \geq -w_0 x_0 \\ 0 & \text{else} \end{cases}$$

- сигмоїдана активаційна функція;

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- гіперболічний тангенс.

$$tg(Ax) = \frac{e^{Ax} - e^{-Ax}}{e^{Ax} + e^{-Ax}}$$

Найбільш відомими моделями штучного нейрону є перцептрон та сигмоїдний нейрон.

Перцептрон - тип штучного нейрона, що розроблений Френком Розенблатом в 1950-их роках. Принцип роботи перцептрону наступний: він приймає на вхід значення x_1, x_2, \dots і видає бінарний результат (див. рис. 2.1).

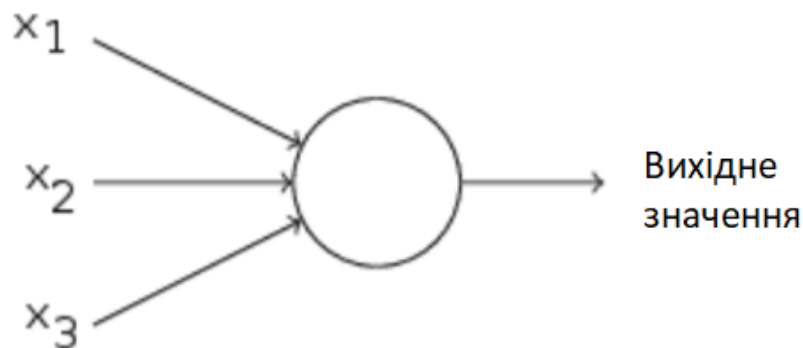


Рисунок 2.1. — Схема роботи перцептрона

Розенблатт запропонував використовувати вагу - числа, що виражають важливість вкладу кожного входу в кінцевий результат. Зважена сума (вага) порівнюється з граничним значенням (threshold) і за результатами визначається: буде виданий 0 або 1. Граничне значення також є параметром нейрона.

$$\begin{cases} 0 \text{ if } \sum_j w_j x_j \leq \text{threshold} \\ 1 \text{ if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Перцептрони можуть бути класифіковані як штучні нейронні мережі

- з одним прихованим шаром;
- з пороговою активаційною функцією;
- з прямим розповсюдженням сигналу.

Навчання перцептрону полягає в корегуванні матриці зважених сум.

Сигмоїдні нейрони схожі на перцептрони, однак невеликі зміни їх ваги та зміщення незначно змінюють результат виходу нейрона. Цей факт дозволяє мережі з сигмоїдних нейронів навчатися. На вхід подаються будь-які значення між 0 і 1. На виході також видається значення між 0 і 1, так як в якості активаційної функції використовується сигмоїда, що є нелінійною (див. Рис. 2.2)

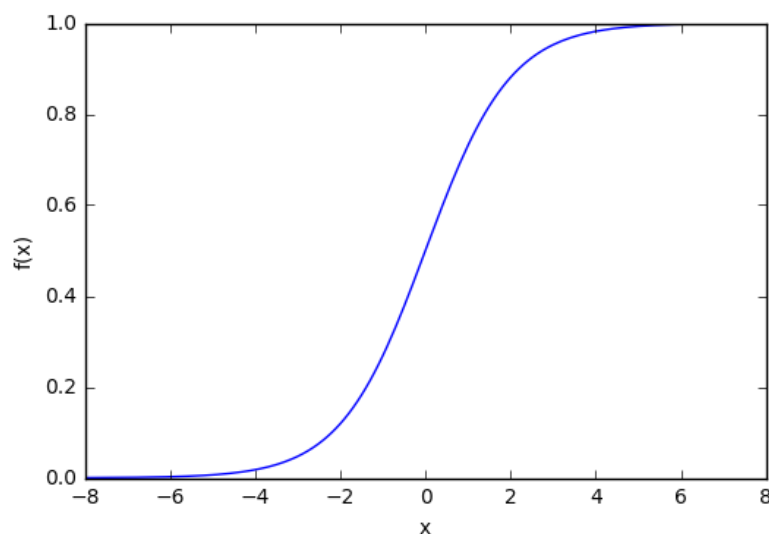


Рисунок 2.2. — Графік сигмоїди

Важливою властивістю сигмоїдної функції є її дифференційність. Застосування неперервної функції активації дозволяє використовувати при навчанні градієнтні методи.

Всі нейронні мережі представлені спрямованим графом певної

архітектури нейронів, які виступають в ролі вершин або вузлів. Нейрони можна розділити на групи в залежності від їх положення в мережі:

- вхідні нейрони (вхідний шар), що приймають певний вектор даних;
- проміжні нейрони (прихований шар), де відбуваються основні обчислювальні операції - навчання;
- вихідні нейрони - результат роботи мережі.

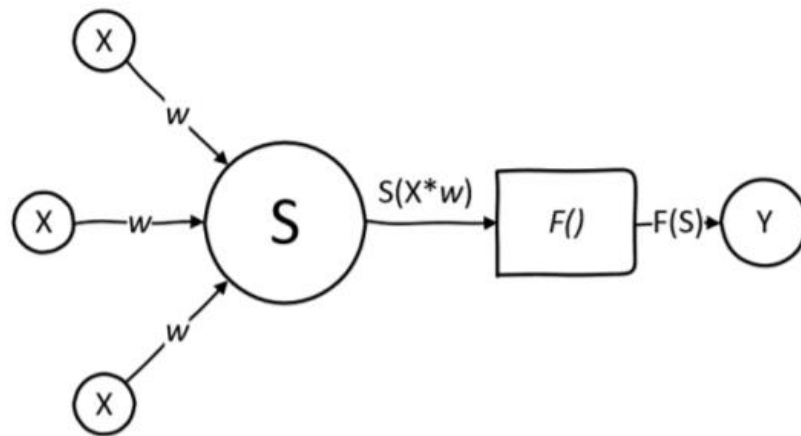


Рисунок 2.3. — Схема нейрона

На рис. 2.3 зображена схема нейрона, де :

- x – входи в нейрон;
- w – вага нейрона;
- S – суматор;
- $F()$ – функція активації;
- Y – вихід нейрона.

Одна з найпростіших нейромережеву систем є систем прямого поширення. Її аналіз наглядно показує роботу нейромережевих систем.

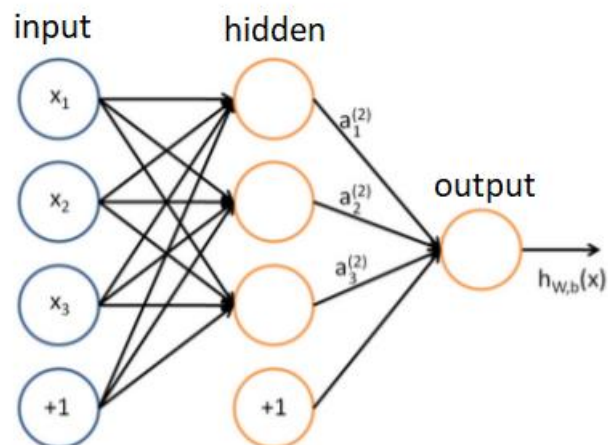


Рисунок 2.4. — Схема простої нейромережі

Нехай n_l - кількість шарів. Параметри мережі – (W,b) , a_i - результат використання функції активації (вихід). Рівняння такої системи:

$$\begin{cases} a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\ \dots \\ a_3^{(2)} \end{cases}$$

У векторній формі :

$$\begin{cases} z^{(2)} = W^{(1)}x + b^{(1)} \\ a^{(2)} = f(z^{(2)}) \\ z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \\ h = f(z^{(3)}) \end{cases}$$

Загальна формула виглядає так:

$$\begin{cases} z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \\ a^{(l+1)} = f(z^{(l+1)}) \end{cases}$$

Мережами прямого поширення називаються нейронні мережі, які використовують вихід одного шару в якості вхідних даних для наступного шару.

Існує значна кількість складніших нейронних мереж порівняно з вище приведеною. Вибір типу нейронної мережі є важливим етапом для застосування її на практиці. Найбільш відомими є:

- багатшаровий персептрон
- згорткові нейромережеві системи (CNN)
- рекурентні нейромережі (RNN)
- мережі довгої короткочасної пам'яті (LSTM)

1. Багатшаровий персептрон має три або більше шари. Він використовує нелінійну функцію активації (в основному гіперболічну), яка дозволяє класифікувати дані, які лінійно не розділяються. Кожен вузол в шарі з'єднується з кожним вузлом на наступному рівні, роблячи мережу повністю підключеною. Використовується, наприклад, для розпізнавання мови і машинного перекладу.

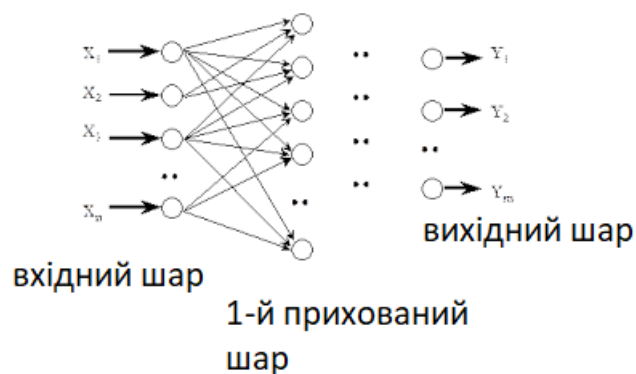


Рисунок 2.5. — Схема багатшарового персептрону

2. Згорткова нейронна мережа (CNN) містить один або кілька згорткових шарів, об'єднаних чи повністю зв'язних, і використовує варіації багатошарових персептронів, розглянутого вище. Шари CNN використовують операцію згортки для вводу результату, що передається на наступний шар. Ця операція дозволяє мережі бути глибше з меншою кількістю параметрів.

Згорткові нейронні мережі демонструють видатні результати в додатках для зображення й мови. Юн Кім в “Згорткових нейронних мережах для класифікації пропозицій” описує процес і результати завдань класифікації тексту з використанням CNN [8]. Він презентує модель, побудовану поверх word2vec, проводить серію експериментів з нею і тестує її, демонструючи, що модель відмінно працює.

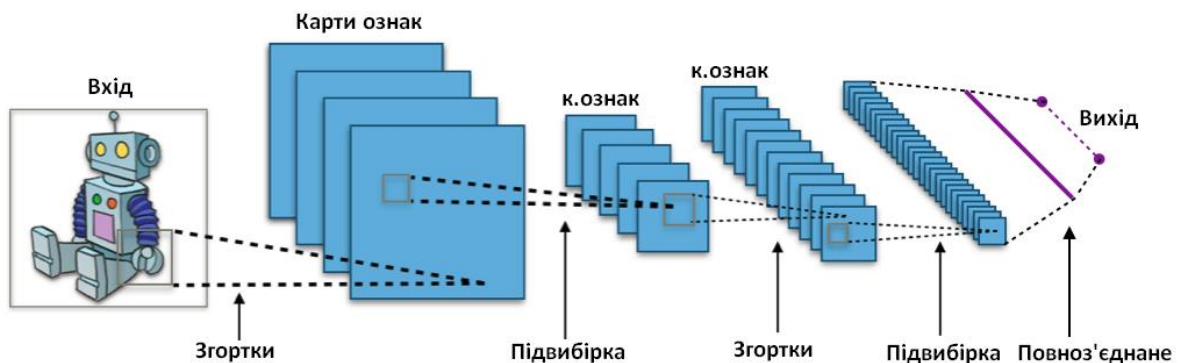


Рисунок 2.6. — Типова структура CNN

3. RNN, на відміну від первинної нейронної мережі, являє собою варіант рекурсивної нейронної мережі, в якій зв'язки між нейронами створюють спрямований цикл. Це означає, що вихід залежить не тільки від поточних входів, а й від стану нейрона попереднього кроку. Ця пам'ять дозволяє користувачам вирішувати проблеми обробки природної мови, такі як розпізнавання рукописного введення або розпізнавання мови.

У статті “Генерація природної мови, зміна і підсумовування

оглядів користувачів з рекурентними нейронними мережами” автори демонструють рекурентну модель нейронної мережі, яка може генерувати нові пропозиції та зведення документів [9].

4. Мережа довгої короткочасної пам'яті являє собою специфічну рекурентну архітектуру нейронної мережі (RNN), яка була розроблена для більш точного моделювання часових послідовностей і їх залежностей, ніж звичайні RNN. LSTM не використовує функцію активації в своїх рекурентних компонентах, збережені значення не змінюються, і не має проблеми зникнення градієнту під час навчання. Зазвичай вузли LSTM реалізовані в “блоках” по кілька одиниць. Ці блоки мають три або чотири “затвора” (вхідний, забуття і вихідний), які керують потоком інформації, використовуючи логістичну функцію.

У статті “Довга короткочасна пам'ять рекурентних нейронних мереж для великомасштабного акустичного моделювання” Х. Сак, Е.Стар і Ф. Бьюфайс показали, що архітектури LSTM рекурентних нейронних мереж забезпечують найвищу продуктивність [10]. Модель досягла продуктивності з відміткою 97.40%. Apple, Amazon, Google, Microsoft та інші компанії включили LSTM в якості основного елемента в свої продукти.

Використовуючи вище приведені дані можна припустити, що рекурентна нейронна мережа з використанням модуля довгої короткочасної пам'яті буде однією з оптимальних для вирішення задачі розпізнавання деструктивного контенту в текстових веб-ресурсах. Потрібно розглянути їх більше детально.

2.3 Використання рекурентних нейронних мереж

Рекурентні нейронні мережі - вид нейронних мереж, в яких сигнали можуть проходити по нейронах як звичайно, так і в зворотний бік. З цією перевагою вони відрізняються від інших мереж прямого поширення. На

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		28

кожному проході сигналу, нейрони запам'ятовують свої стани і використовують попередні стани мережі, для обчислення поточного [11]. За аналогією з людським мозком, можна сказати, що рекурентні нейронні мережі додають функцію пам'яті до штучних НМ. Цей клас нейронних мереж зарекомендував себе в задачах обробки послідовностей, де їх довжина не є фіксованою. Тобто, на відміну від інших мереж, де вибірки повинні бути представлені у вигляді фіксованого за величиною набору даних, RNN не прив'язані до величини навчального екземпляру, тому вони показують кращі результати по співвідношенню до інших методів в задачах класифікації тексту.

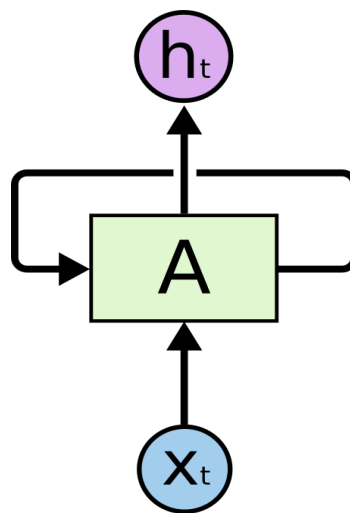


Рисунок 2.7. — Архітектура найпростішої рекурентної нейромережі

RNN містять у собі зворотні зв'язки. На рис. 2.7. фрагмент А нейронної мережі приймає певне входнє значення і повертає вихідне. Наявність зворотного зв'язку дозволяє передавати інформацію в мережі покроково.

На кожному кроці t навчання системи значення нейронів в прихованому шарі нейронної мережі обчислюється за формулою.

$$k_t = f(Wx_t + Uk_{t-1} + bk)$$

В данній формулі x_t – вхідний вектор в момент часу t ; W , U , b_k – параметри нейронної мережі, які обчислюються при навчанні; f – функція активації.

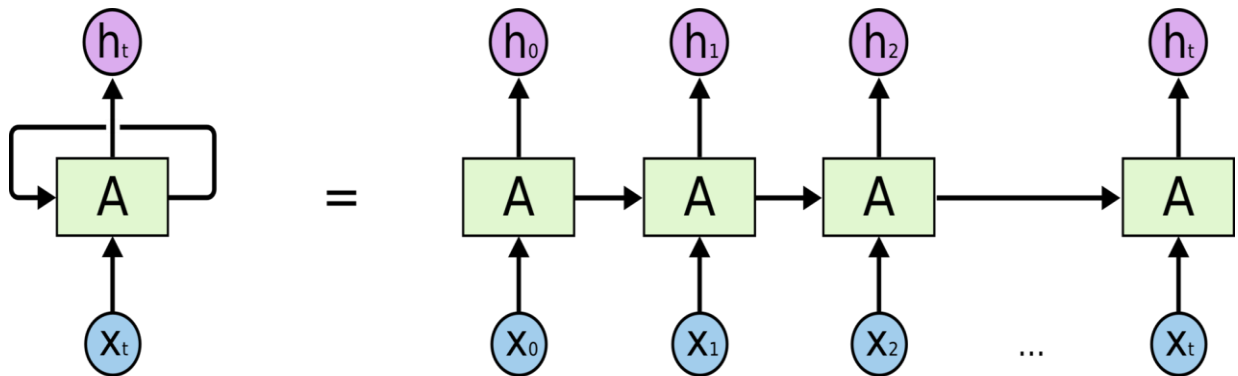


Рисунок 2.8. — Архітектура найпростішої рекурентної нейромережі при кількості t ітерацій роботи

В випадку простої рекурентної нейронної мережі, яка зображена на рис. 2.8. , ми обчислюємо вихідне значення h_t на кроці t так:

$$h_t = Wc_k t + b_c$$

Однак та ж сама пам'ять, в даному випадку архітектурної реалізації, є дуже короткою, звідки впливає наступна проблема. В процесі навчання нові вхідні дані, які мережа має опрацювати починають впливати на пам'ять мережі: інформація в пам'яті змішується з новою інформацією та через декілька ітерацій повністю перезаписується.

Для задач аналізу тональності тексту це особливо важливо, адже розглядаючи великий текстовий корпус, існує ризик побудувати навчання лише на його останньому уривку. Адже слова, що стоять на початку, не будуть вносити зміни в нейронні ваги і будь-яким чином враховуватися в результатах роботи нейронної мережі ближче до кінця тексту. В такому випадку помилка на перших словах тексту або речення буде сильно

впливати на загальну помилку нейронної мережі.

Таке явище називається проблемою зникаючого градієнта (vanishing gradient problem). Цікавим є те, що в певний період часу вона навіть була причиною застою в дослідженнях нейронних мереж - дослідники переключилися на метод опорних векторів. Ця проблема характерна для обох видів нейронних мереж: мереж прямого поширення і нейронних мереж із зворотними зв'язками.

Незважаючи на те, що іноді вдається знайти конкретне рішення при певній задачі та архітектури мережі, але якогось універсального рішення для усунення проблеми зникаючого градієнта немає, так що вона й по сьогоднішній день залишається актуальною. Робота [12] повністю присвячена проблемі зникаючого градієнта в нейронних мережах зворотного поширення сигналу.

Навчання рекурентних нейронних мереж здійснюється за допомогою алгоритму зворотного поширення помилки за часом (backpropagation through time) [13], який є давно відомим алгоритмом зворотного поширення помилки, що використовується для навчання нейронних мереж, побудованих за принципом прямого поширення сигналу. Реалізація даного алгоритму проілюстрована на рис. 2.9. за допомогою псевдокоду. Мережа навчається до тих пір, поки не спрацює умова зупинки процесу. Ця умова спрацьовує в разі, якщо нейронна мережа навчається задану максимальну кількість ітерацій або в разі, коли застосовується метод ранньої зупинки. Суть методу полягає в тому, що після певної кількості ітерацій навчання вводиться додатковий набір даних для перевірки, який нейронна мережа класифікує після кожної ітерації навчання. Тобто, щось на зразок мінімального тестового набору для перевірки роботи мережі. Тим самим, алгоритм дозволяє відстежити, коли результати класифікації погіршуються або не змінюються протягом кількох ітерацій та припинити навчання.

```

Back_Propagation_Through_Time(a, y)  // a[t] is the input at time t. y[t] is the output
    Unfold the network to contain k instances of f
    do until stopping criteria is met:
        x = the zero-magnitude vector; // x is the current context
        for t from 0 to n - k          // t is time. n is the length of the training sequence
            Set the network inputs to x, a[t], a[t+1], ..., a[t+k-1]
            p = forward-propagate the inputs over the whole unfolded network
            e = y[t+k] - p;              // error = target - prediction
            Back-propagate the error, e, back across the whole unfolded network
            Sum the weight changes in the k instances of f together.
            Update all the weights in f and g.
        x = f(x, a[t]);                 // compute the context for the next time-step

```

Рисунок 2.9. — Псевдокод роботи алгоритму зворотного поширення помилки за часом

Оскільки алгоритм зворотнього поширення помилки за часом використовує в своїй основі звичайний метод зворотного поширення помилки, в ньому застосовуються ті ж методи для боротьби з проблемою перенавчання нейронних мереж.

В залежності від сформованого входу та виходу рекурентної нейронної мережі [18], схема роботи RNN відрізняється одна від одної.

Можливі способи організації роботи рекурентної мережі [18]:

- many-to-one (багато входів, один вихід) - прихований шар послідовно змінює свій стан, з його кінцевого стану обчислюється вихід мережі. Цю схему доцільно використовувати для класифікації текстів (див рис. 2.10.);
- one-to-many (див рис. 2.11.) - прихований шар ініціалізується одним входом, з ланцюжка його наступних станів генеруються виходи мережі, цю схему можна використовувати для пошуку підпису до зображення;
- many-to-many (див рис. 2.12.) - на кожен вхід мережа видає вихід,

який залежить від попередніх входів, цю схему можна використовувати для класифікації відео;

- many-to-many - прихований шар послідовно змінює свій стан, його кінцевий стан служить ініціалізацією для видачі ланцюжка результатів. Цю схему можна використовувати для створення систем машинного перекладу і чат-ботів (див рис. 2.13.).

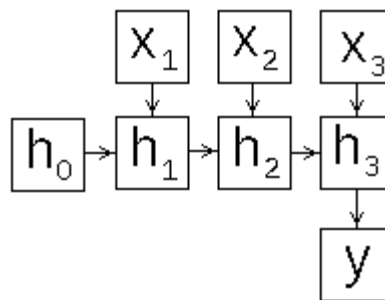


Рисунок 2.10. — Схема роботи many-to-one RNN мережі

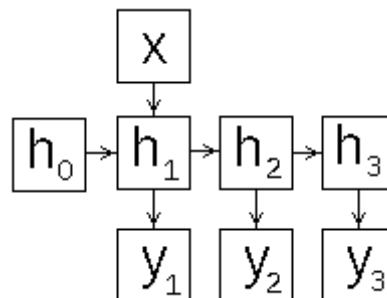


Рисунок 2.11. — Схема роботи one-to-many RNN мережі

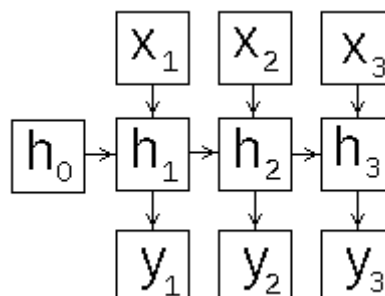


Рисунок 2.12. — Схема роботи many-to-many RNN мережі

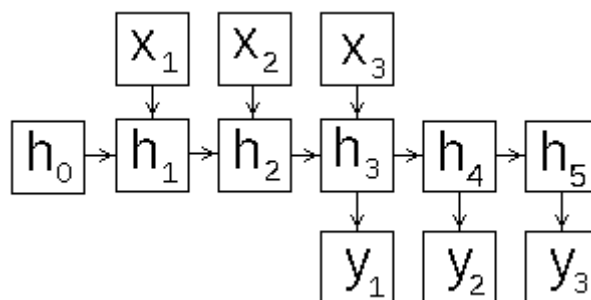


Рисунок 2.13. — Схема роботи many-to-many RNN мережі

З появою проблеми зникаючого градієнта в рекурентних нейронних мережах перспективним завданням став пошук її рішення. Була представлена архітектура довгої короткочасної пам'яті, яка успішно зарекомендувала себе [14]. LSTM (long short-term memory) – це різновид рекурентних нейронних мереж, яка дає змогу довше зберігати ті ваги в нейронах, що відповідають за початкові вхідні дані, та забороняє повністю їх перезаписати. На рис. 2.14. зображена схема такої мережі. Структура LSTM також, як і RNN нагадує ланцюжок, але модулі виглядають інакше. Замість одного шару нейронної мережі вони містять цілих чотири, і ці шари взаємодіють особливим чином.

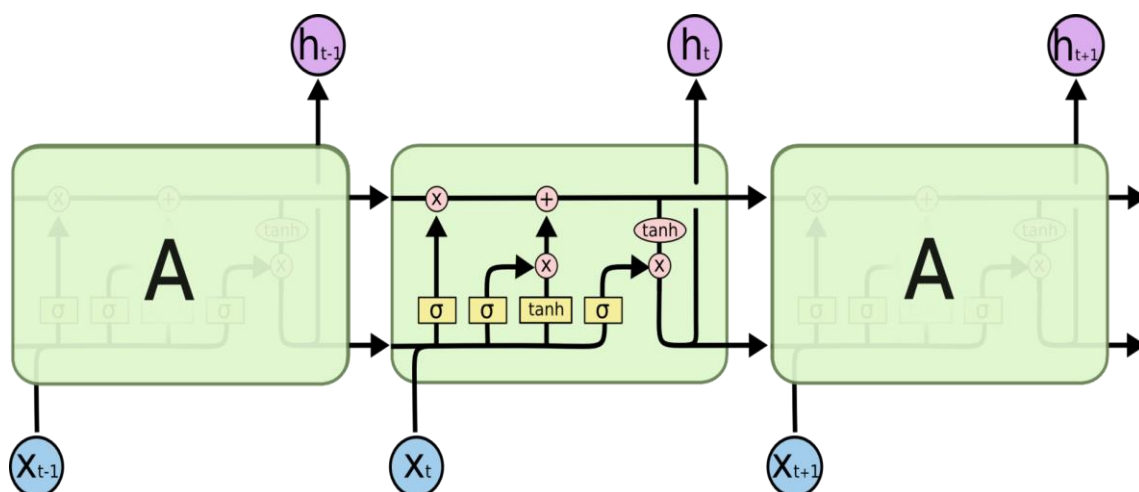


Рисунок 2.14. — Архітектура довгої короткочасної пам'яті в певний період часу t

Модуль, який повторюється в LSTM мережі складається з чотирьох взаємодіючих шарів. В ньому відбуваються операції складання векторів, копіювання даних в різні компоненти мережі. Звичайно, модуль містить навчені шари нейромережі.

Ключовим компонентом мережі довгої короткочасної пам'яті є стан комірки. Інформація по ній може вільно проходити без ризику змін. Комірка задієна тільки в деяких лінійних перетвореннях. Проте LSTM може видаляти інформацію зі стану комірки, що під контролем особливих структур – фільтрів. На підставі певних умов, фільтри дають змогу пропускати інформацію. Вони складаються (див. рис. 2.15.) з сигмоїдного шару та операції точкового множення. Сигмоїдний шар повертає числа від нуля до одиниці. Значення чисел показують, який процент інформації кожного блоку слід пропустити далі по мережі. Нуль в даному випадку означає - не пропускати нічого, одиниця - пропустити всі дані.

У LSTM три таких фільтра, які дозволяють захищати і контролювати стан комірки.

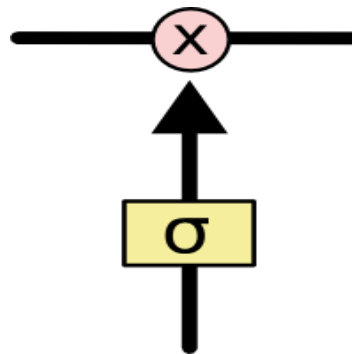


Рисунок 2.15. — Архітектура фільтра LSTM

В загальних випадках модуль LSTM складається з чотирьох основних шарів:

- вхідний шар;

- шар забуття інформації;
- шар збереження інформації;
- вихідний шар.

Шар забуття (forget gate layer) визначає, яку інформацію можна викинути зі стану комірки. Він аналізує значення h_{t-1} , x_t та повертає число від 0 до 1 для кожного числа стану комірки C_{t-1} . Одиниця означає повністю зберегти, 0 – викинути всі дані.

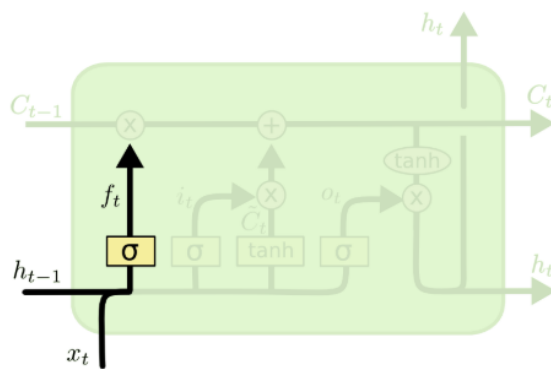


Рисунок 2.16. — Шар забуття інформації

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

Збереження нової інформації в стан комірки відбувається в два етапи. Спочатку вхідний шар (input layer gate) визначає значення, які потрібно оновити. Пізніше шар tanh будує вектор нових можливих значень C'_t , що можна додати в стан комірки.

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

При видаленні старої інформації та додавання нової - стан комірки з C_{t-1}

змінюється на C_t :

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

До вихідних даних, заснованих на стані комірки, застосовуються деякі фільтри. Спочатку використовується шар забуття. Потім значення стану комірки проходить через шар \tanh , щоб отримати на виході значення в діапазоні від -1 до 1, і перемножуються з вихідними значеннями сигмоїдного шару, що дозволяє виводити тільки необхідну інформацію.

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

Для продовження навчання нейронної мережі використовується алгоритм зворотнього розповсюдження помилки за часом.

Не всі мережі довгої короткочасної пам'яті однакові. Взагалі, здається, що в кожній новій роботі, присвяченій LSTM, використовується своя версія LSTM. Відмінності між ними незначні, але про деякі з них варто згадати.

Одна з популярних варіацій цих нейронних мереж, запропонована Герсом і Шмідхубером в [15], характеризується додаванням так званих "peerhole connections". З їх допомогою шари фільтрів можуть бачити стан комірки (див. рис. 2.17.).

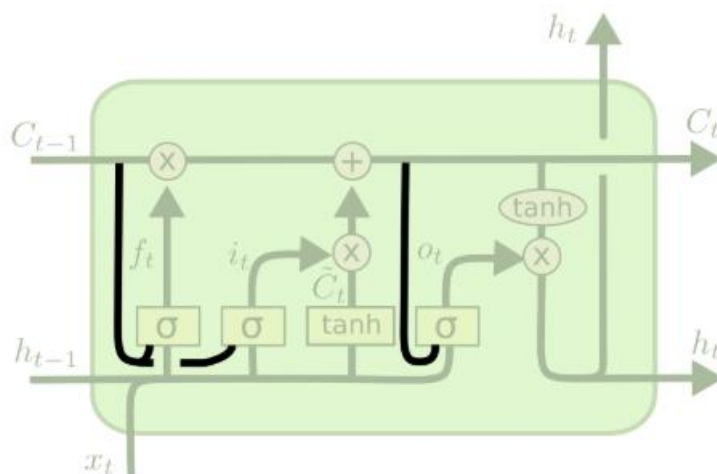


Рисунок 2.17. — Схема варіації LSTM

$$f_t = \sigma (W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma (W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma (W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

На схемі Герса “peerhole connections” є у кожного шару, але в багатьох роботах вони додаються лише до деяких шарів.

Альтернативним варіантом модифікації є об’єднання вхідних фільтрів та фільтрів забуття. У цьому випадку рішення щодо інформації, яку слід забути, а яку запам’ятати - приймаються не окремо, а спільно. Забуття інформації виконується тоді, коли необхідно записати щось на її місце. Додавання ж нових даних в стан комірки проходить при забутті старих.

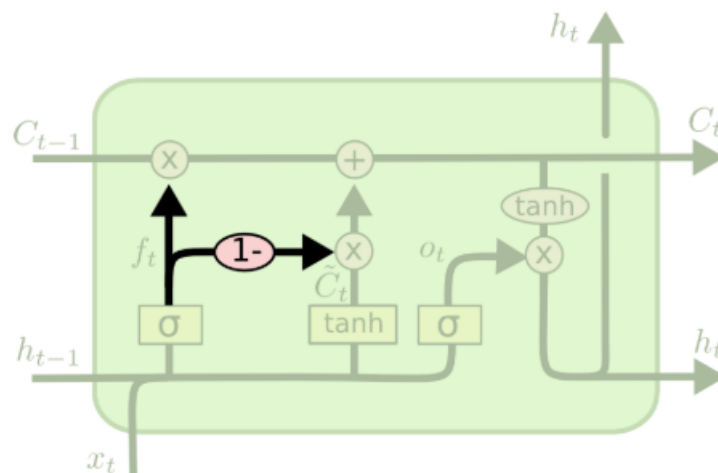


Рисунок 2.18. — Схема варіації LSTM

$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

Ще однією варіацією рекурентної мережі є архітектура GRU (Gated Recurrent Unit) або “Керований рекурентний блок” [16]. Розглянемо цей вид нейронної мережі. Представлена в 2014 році GRU була ще однією спробою вирішити проблему зникаючого градієнта, з якою стикається кожен алгоритм класифікації, побудований на стандартній рекурентній нейронної мережі. В цілому, GRU можна розглядати як варіант мережі LSTM, оскільки вони розроблені аналогічним чином і в деяких більшості випадків дають однакові результати.

У розробці GRU за основу взяли звичайну рекурентну мережу, але додали між її шарами кілька особливостей, які впливають на кожну ітерацію навчання мережі. Перше нововведення це два вектора, котрі називаються update gate і reset gate. Вони є своєрідним фільтром, який визначає, яка інформація повинна бути передана на вихід.

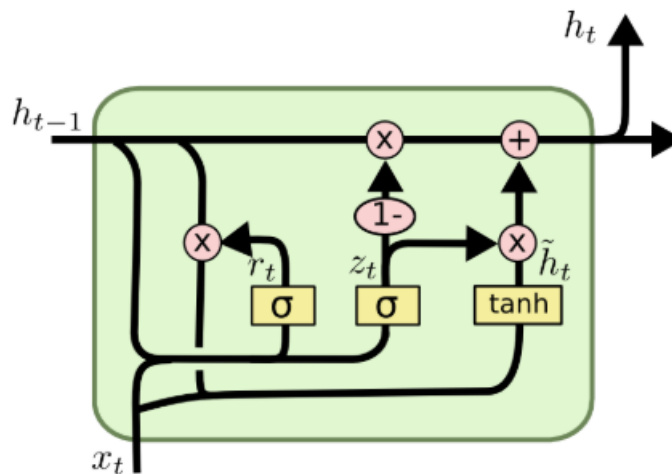


Рисунок 2.19. — Архітектура GRU мережі

$$z_t = \sigma (W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma (W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh (W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Основна особливість цих векторів, це їх здатність навчатися і розуміти, яку інформацію треба пропускати в пам'ять, а яку варто видалити. Тобто, вони дозволяють зберігати інформацію про перші входні дані довше ніж LSTM мережі, якщо вона несе в собі користь.

Перша формула показує, як обчислюється update gate (z_t) для певного кроку t . Коли x_t підключається до блоку мережі, він множиться на власну вагу $W(z)$. Те ж саме відбувається з h_{t-1} , який містить інформацію від попередньої ітерації і множиться на власну вагу $U(z)$.

Потім, обидва результати складаються разом, і сигмоїдна функція активації стискає отриманий сигнал в діапазоні від 0 до 1.

Update gate вектора допомагають моделі визначити, яка частина інформації (з попередніх кроків) повинна бути передана далі. Це дуже ефективний інструмент, тому що модель може вирішити зовсім не міняти інформацію з минулого ітерації, тим самим повністю усунути ризик зникнення проблеми градієнта.

У свою чергу, reset gate (r_t) вектора навпаки, вирішують, яку інформацію з минулого ітерації слід забути.

Після обчислення значень reset та update gate – визначається поточний стан пам'яті. Для цього потрібно з'ясувати, який саме вектор вплинув на кінцевий результат.

В якості останнього кроку мережі необхідно обчислити h_t - вектор, який містить інформацію вузла і передає її на вихід до наступного вузла мережі.

3. РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ

3.1 Модель рекурентної нейромережі для класифікації тексту

Для вирішення проблеми перенавчання мережі використовується техніка dropout, докладніше описана в роботі [19]. Сенс в тому, що при роботі з деякою випадковою величиною (зазвичай, розподіленою за законом Бернуллі) на входи наступного шару, які не отримали сигнал із-за того, що значення у вхідній базі закінчилися - надходить “0”, а не значення, яке було передано з попереднього шару. В даному методі використовується шар dropout з випадковою величиною, розподіленою за законом Бернуллі, при ймовірності настання події 0.5.

Спочатку попередньо оброблені дані, у вигляді масивів чисел (токенів), надходять через вхідний шар до Embedding шару (векторного представлення слів). Цей шар має два параметра: максимальна кількість слів в тексті та задану кількість вимірів. За допомогою його функціонування спрощується та прискорюється навчання мережі в цілому, так як буде необхідна менша кількість вхідних нейронів, яка буде дорівнювати заданій кількості вимірів. При цьому Embedding шар кожному слову призначає вектор ваг, який необхідно вивчити в процесі навчання мережі. Тобто кожний вхід (тензор з k словами) перетворюється в k N -мірний тензор векторного представлення слів (N – задана кількість вимірів). Після цього дані, у вигляді векторів надходять до LSTM шару, де виконуються певні операції над ними. LSTM шар визначається прихованими значеннями стану та певною кількістю шарів.

Наступним, перед вихідним, знаходиться повнозв'язний шар. Він відображає вихідні дані LSTM та бажаний вихідний результат (у нашому випадку 1). Для класифікації тексту використовується сигмоїдна активаційна функція. Результатом її роботи буде число в інтервалі $[0,1]$. Вихідний сигнал сигмоїда з останнього часового кроку рахується кінцевим

результатом роботи мережі, який видає вихідний шар. Оскільки в нашому випадку класифікація бінарна, то значення менші 0.5 означають, що текст є негативним, в іншому випадку він відноситься до позитивного.

Отже, архітектура нашої моделі рекурентної мережі складається з наступних компонентів:

1. Вхідний шар.
2. Embedding шар.
3. Шар LSTM з використанням dropout.
4. Повнозв'язний шар.
5. Вихідний шар.

3.2 Програмні засоби для роботи системи

Описана архітектура рекурентної нейронної мережі була реалізована на мові програмування Python 3.7. Вона дуже популярна в області машинного навчання та містить значну кількість бібліотек для роботи з нейронними мережами. Була обрана бібліотека Keras, яка виникла з не менш популярної - Theano. Дана бібліотека полегшує роботу з обчисленнями, дозволяючи використовувати як CPU (центральний процесор), так і GPU (графічний процесор), в якому реалізована технологія Nvidia CUDA. Найчастіше розрахунки на GPU в завданнях машинного навчання гарантують високу продуктивність. В роботі [20] демонструється перевага GPU над CPU, які справляються з такими завданнями швидше в кілька разів.

В якості бекенда, окрім Keras, для обчислень скористаємося бібліотекою TensorFlow. Так як необхідний високопродуктивний комп'ютер використаємо хмарну платформу Colaboratory, розробленою Google. В ній встановлені бібліотеки TensorFlow і Keras, а також доступний GPU NVIDIA K80.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		43

Вхідною базою даних для навчання та тестування мережі є загальнодоступна база рецензій imdb [21], яка вже вбудована в бібліотеці Keras.

Реалізацію потрібної нейромережевої системи можна розбити на три модулі:

- попередня обробка даних;
- створення моделі та навчання рекурентної нейронної мережі;
- проведення експериментів.

3.2.1 Попередня обробка даних

Даний крок необхідний для того, щоб підготувати вхідні дані до навчання мережі і тестування її здатності класифікувати їх. Для цього текст очищається від символів, які не впливають на класифікацію, тобто не несуть смислового навантаження в завданні аналізу тональності. Зрозуміло, що посилання у відгуку не мають емоційного забарвлення, хоча і можуть впливати на зміст. Так як ми працюємо з англійською мовою необхідно перетворити негативні форми: замінити “are not” на “are not” і т.д.

Всі теги та символи пунктуації також повинні бути видалені. Вони можуть негативно вплинути на навчання мережі, так як володіють значним, зайвим текстовим обсягом.

Для зручності переведемо всі слова в нижній регістр. Він також не дає нам інформації про тональність тексту.

Необхідним кроком є видалення з бази прийменників та артиклів, які не несуть емоційного забарвлення, а є правилами правопису. Без ризику втратити цінну для тональності інформацію можна прибрати всі прості слова з таблиці 3.1.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		44

Таблиця 3.1

Прості прийменники з англійської мови

aboard	in	save	before	at	via
about	inside	since	behind	aside	vice
above	into	than	below	despite	with
absent	like	through	beneath	bar	without
across	mid	till	beside	before	post
afore	minus	times	besides	pro	out
after	near	to	between	qua	over
against	neath	towards	versus	round	pace
along	next	under	beyond	down	per
amid	of	unlike	but	except	plus
amidst	off	until	by	for	around
among	on	up	opposite	from	as

Після цього потрібно створити словник слів, а речення замінити на списки чисел, які дорівнюють номерам слів в словнику. Також необхідно встановити максимальну довжину тексту (списка чисел), та використовувати її в якості довжини вхідного вектора слів.

Всі операції для попередньої обробки:

- видалення з тексту символів пунктуації;
- видалення посилань та тегів;
- приведення в нижній регістр;
- перетворення негативних форм;
- видалення артиклів і прийменників з тексту;
- створення словника слів;
- заміна кожного слова в реченні числом;
- приведення всіх векторів чисел до одного розміру.

4. АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Технологія використання

Базова версія програми реалізує навчання нейронної мережі для визначення тональності рецензій до фільмів. Повна версія системи на основі визначеної тональності - проводить аналіз щодо деструктивності тексту нових відгуків. Програма працює в режимі ноутбука з використанням хмарного середовища Colaboratory.

В якості умови зупинки навчання мережі була обрана модифікація методу ранньої зупинки, який припиняє навчання, якщо результати залишалися незмінними або ставали лише гірше на протязі 5 ітерацій навчання.

У процесі роботи з вибіркою та реалізації програми скористаємося ефективним алгоритм оптимізації Adam (adaptive moment estimation) [22]. Потім спробуємо замінити його на стандартний алгоритм стохастичного градієнтного спуску (Stochastic Gradient descent). З бібліотеки Keras були імпортовані такі компоненти: шари LSTM, Dense, Embedding, активаційна функція та модуль sequence для роботи з даними.

У базовій версії програми, були визначені такі параметри:

1. Об'єм словника – 5000 слів.
2. Довжина рецензій у словах – 100.
3. Кількість епох навчання – 7.
4. Кількість нейронів в шарі LSTM – 100.
5. Довжина вектора embedding – 32.

Якість роботи мережі можна побачити на рис. 3.1., де показана точність системи на тестових даних – 83.84%.

```

Train on 25000 samples, validate on 25000 samples
Epoch 1/7
25000/25000 [=====] - 108s 4ms/step - loss: 0.4621 - acc: 0.7728 - val_loss: 0.3680 - val_acc: 0.8405
Epoch 2/7
25000/25000 [=====] - 106s 4ms/step - loss: 0.3143 - acc: 0.8698 - val_loss: 0.3586 - val_acc: 0.8444
Epoch 3/7
25000/25000 [=====] - 107s 4ms/step - loss: 0.2790 - acc: 0.8857 - val_loss: 0.3427 - val_acc: 0.8501
Epoch 4/7
25000/25000 [=====] - 106s 4ms/step - loss: 0.2510 - acc: 0.8996 - val_loss: 0.4415 - val_acc: 0.8376
Epoch 5/7
25000/25000 [=====] - 106s 4ms/step - loss: 0.2256 - acc: 0.9109 - val_loss: 0.3952 - val_acc: 0.8453
Epoch 6/7
25000/25000 [=====] - 106s 4ms/step - loss: 0.2013 - acc: 0.9226 - val_loss: 0.4065 - val_acc: 0.8436
Epoch 7/7
25000/25000 [=====] - 106s 4ms/step - loss: 0.1836 - acc: 0.9294 - val_loss: 0.3981 - val_acc: 0.8404
Accuracy: 84.04%

```

Рисунок 3.1. — Процес навчання мережі на тестових даних

```

=====
i really liked the movie and had fun . Sentiment: 0.90511465
Text is not destructive
=====
=====
this movie was terrible and bad . Sentiment: 0.05767885
Text is destructive
=====

```

Рисунок 3.2. — Результат роботи створеної системи

На рис. 3.2 зображений результат функціонування системи при вводі нових даних.

Реалізація рекурентної нейронної мережі складається з таких кроків:

- етап попередньої обробки даних;
- збереження множини відомих слів;
- перетворення навчальної вибірки в векторне подання “bag of words” на основі множини відомих слів;
- навчання рекурентної нейронної мережі методом зворотного поширення помилки за часом.

Класифікація нового тексту поділяється на етапи:

- етап попередньої обробки даних;
- перетворення тексту в вигляді “bag of words” на основі множини

відомих слів;

- поширення сигналів кожного слова один за одним по нейромережі;
- класифікація тексту за значенням вихідного нейрона. Якщо воно менше 0.5, то текст відноситься до негативного класу в іншому випадку – до позитивного;
- проведення оцінки деструктивності тексту, в залежності від визначеного класу.

4.2 Експериментальні дослідження

Ми можемо спробувати покращити якість навчання мережі шляхом зміни її параметрів таких як: кількість епох навчання, алгоритм оптимізації, кількість нейронів в LSTM шарі. Зробимо серію додаткових експериментів, в кожному з яких будемо змінювати один з параметрів і аналізувати нові результати роботи мережі.

Для початку оцінимо вплив кількості епох на якість навчання мережі. Стандартно мережа навчається сім раз. Насправді RNN вчатья дуже швидко і здатні навіть з першої спроби видавати високу точність. Проте, спробуємо навчати нейромережу протягом 5, 10 і 15 епох, для того щоб визначити, коли починається процес перенавчання. Таким чином, легко побачити при якій кількості даного параметру - найвища точність роботи LSTM на тестових даних. Результати експериментів виведені в таблицю 4.1.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		48

Таблиця 4.1

Залежність точності класифікації від кількості епох

Кількість епох	Точність
5	83.31%
10	82.51%
15	82.50%

Як видно з нашої таблиці, алгоритм зупинки працює вірно, адже на сьомій епосі навчання ми отримуємо найвищу точність. Вона все ще висока і на п'ятій епосі з 83,31%, але потім мережа починає перенавчатися і точність падає на десятій епосі і ще більше на п'ятнадцятій. Логічно, адже рекурентні мережі навчаються швидко і навіть на першій кроці ми спостерігали ефективність роботи близьку до п'ятнадцятого.

Тепер, спробуємо змінити кількість нейронів в шарі LSTM. Оцінимо вплив їх кількості на якість навчання. Для наочності вибірки візьмемо 50, 100 та 150 нейронів. Проаналізуємо кількість нейронів в LSTM шарі, при якому спостерігається найбільшу точність. Таке є можливість визначити вплив їх числа на час навчання мережі.

Таблиця 4.2

Залежність точності та часу навчання від кількості нейронів

Кількість нейронів	Час навчання	Точність
50	44s	83.63%
100	106s	84.04%
150	145s	84.34%

Проаналізувавши дані, можна побачити – чим більша кількість використаних нейронів, тим більша точність оцінки та довший час

навчання. Для скорочення часу навчання мережі раціонально використовувати GPU.

При використанні стандартного алгоритму оптимізації стохастичного градієнтного спуску точність класифікації падає до 63.66%.

Звідси можна зробити висновок, що для досягнення найбільшої ефективності функціонування нашої мережі слід використовувати: сім епох навчання, алгоритм оптимізації adam та GPU зі збільшенням кількості нейронів до 300-400.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		50

ВИСНОВКИ

Даний дипломний проект присвячений розв'язанню задачі розробки нейромережевої системи розпізнавання деструктивного контенту в текстових веб-ресурсах. В процесі розв'язання отримано наступні результати:

1. Показано, що для більш ефективного розпізнавання деструктивного контенту в текстових веб-ресурсах необхідно вдосконалити математичне та інформаційне забезпечення системи. В системі розпізнавання перспективно використовувати сучасну модифікацію рекурентної нейронної мережі, яка проводить аналіз тональності тексту.

2. Розроблена структура та математичне забезпечення рекурентної нейронної мережі типу LSTM, що пристосована до класифікації тексту в веб-ресурсах.

3. Розроблена архітектура нейромережевої системи розпізнавання, яка застосовує розроблене математичне забезпечення.

4. Спроектовано програмне забезпечення системи розпізнавання деструктивного контенту, що базується на аналізі тональності тексту з використанням рекурентної нейронної мережі.

5. Проведені експериментальні дослідження, спрямовані на визначення ефективності розпізнавання деструктивного контенту за допомогою побудованої моделі RNN. Були експерименти для порівняння точності роботи мережі у зв'язку зміни внутрішніх параметрів. Результатами експериментів визначено найефективнішу модель з необхідними характеристиками.

6. Розроблену нейромережеву систему рекомендується використовувати в складі інформаційних систем для пошуку відгуків з певним емоційним забарвленням та їх класифікації.

					ІАЛЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		51

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Злоказов, К.В. Контент анализ текстов деструктивной направленности [Электронный ресурс] / К.В. Злоказов. Политическая лингвистика, 2015 — Режим доступа: <http://elar.uspu.ru/bitstream/uspu/1598/1/plin-2015-36.pdf> — Дата доступа: травень 2019.
2. Обучаем компьютер чувствам [Электронный ресурс] / Александр Пак. Анализ тональности, 2012 — Режим доступа: <https://habr.com/ru/post/149605> — Дата доступа: травень 2019.
3. Bo Pang, Lillian Lee. Opinion Mining and Sentiment Analysis [Text] / Foundations and Trends in Information Retrieval. — 2008. — Vol. 2, № 2. — P. 1-135.
4. М.В. Клековкина, Е.В. Котельников. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики [Текст] / RCDL-2012, Переславль-Залесский, Россия. — 2012. — 81 с.
5. И. Меньшиков. Анализ тональности текста на русском языке при помощи графовых моделей [Текст] / УРФУ, Екатеринбург, Россия. — 2012. — с. 1-4.
6. Nozomi Kobayashi, Ryu Iida, Kentaro Inui, Yuji Matsumoto. Opinion Mining on the Web by Extracting Subject-Aspect-Evaluation Relations [Text] / Nara Institute of Science and Technology, Takayama. — 2006. — P. 1-6.
7. Ogneva, M. How Companies Can Use Sentiment Analysis to Improve Their Business [Electronic resource] / Mashable, 2010 — Mode of access: <https://mashable.com/2010/04/19/sentiment-analysis/> — Last access: 2019.
8. Convolutional Neural Networks for Sentence Classification [Electronic resource] / Yoon Kim, 2014 — Mode of access: <http://www.aclweb.org/anthology/D14-1181> — Last access: 2019.
9. Part-of-Speech Tagging with Bidirectional Long Short-Term Memory

Recurrent Neural Network [Electronic resource] / Peilu Wang, Yao Qian, Frank K. Soong, Lei He, Hai Zhao, 2015 — Mode of access: <https://arxiv.org/pdf/1510.06168.pdf> — Last access: 2019.

10. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling [Electronic resource] / H. Sak, A. Senior, F. Beaufays. Google, 2014 — Mode of access: <https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/ListenTerm1201415/sak2.pdf> — Last access: 2019.

11. Elman J.L. Finding structure in time [Text] / Cognitive science. — 1990. — Vol. 14. — P. 179 – 211.

12. Hochreiter S., Bengio Y., Frasconi P., Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. [Electronic resource] / A Field Guide to Dynamical Recurrent Neural Networks. IEEE press, 2001 — Mode of access: <ftp://ftp.idsia.ch/pub/juergen/gradientflow.pdf> — Last access: 2019.

13. Werbos P.J. Backpropagation through time: what it does and how to do it. [Text] / Proceedings of the IEEE 78, 1990 — Issue 10. — P. 1550 – 1560.

14 Hochreiter S., Schmidhuber J. Long short-term memory [Text] / Neural computation 9, 1997 — Issue 8. — P. 1735 – 1780.

15. Recurrent Nets that Time and Count [Electronic resource] / Felix A. Gers, Jurgen Schmidhuber, 2000 — Mode of access: <ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf> — Last access: 2019.

16 Understanding GRU Networks [Electronic resource] / Simeon Kostadinov, Towards Data Science, 2017 — Mode of access: <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be> — Last access: 2019.

17. Jeffrey L. Elman Finding Structure in Time [Text] / Cognitive science 14. – 1990. – P. 179-211.

					ІАЛІЦ.045420.004 ПЗ	Арк.
Змін.	Лист	№ докум.	Підпис	Дата		53

18. The Unreasonable Effectiveness of Recurrent Neural Networks [Electronic resource] / Andrej Karpathy blog, 2015 — Mode of access: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/> — Last access: 2019.
19. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting [Text] / The Journal of Machine Learning Research 15, Harvard. – 2014. – Issue 1. – P. 1929-1958.
20. Oh K.S., Jung K. GPU implementation of neural networks [Text] / Pattern Recognition 37. – 2004. – Issue 6. – P. 1311-1314.
21. Large Movie Review Dataset [Electronic resource] / Stanford Artificial Intelligence Laboratory, 2011 — Mode of access: <http://ai.stanford.edu/~amaas/data/sentiment/> — Last access: 2019.
22. Методы оптимизации нейронных сетей [Электронный ресурс] / Павел Садовников. Adam, 2017 — Режим доступа: <https://habr.com/ru/post/318970/> — Дата доступа: травень 2019.